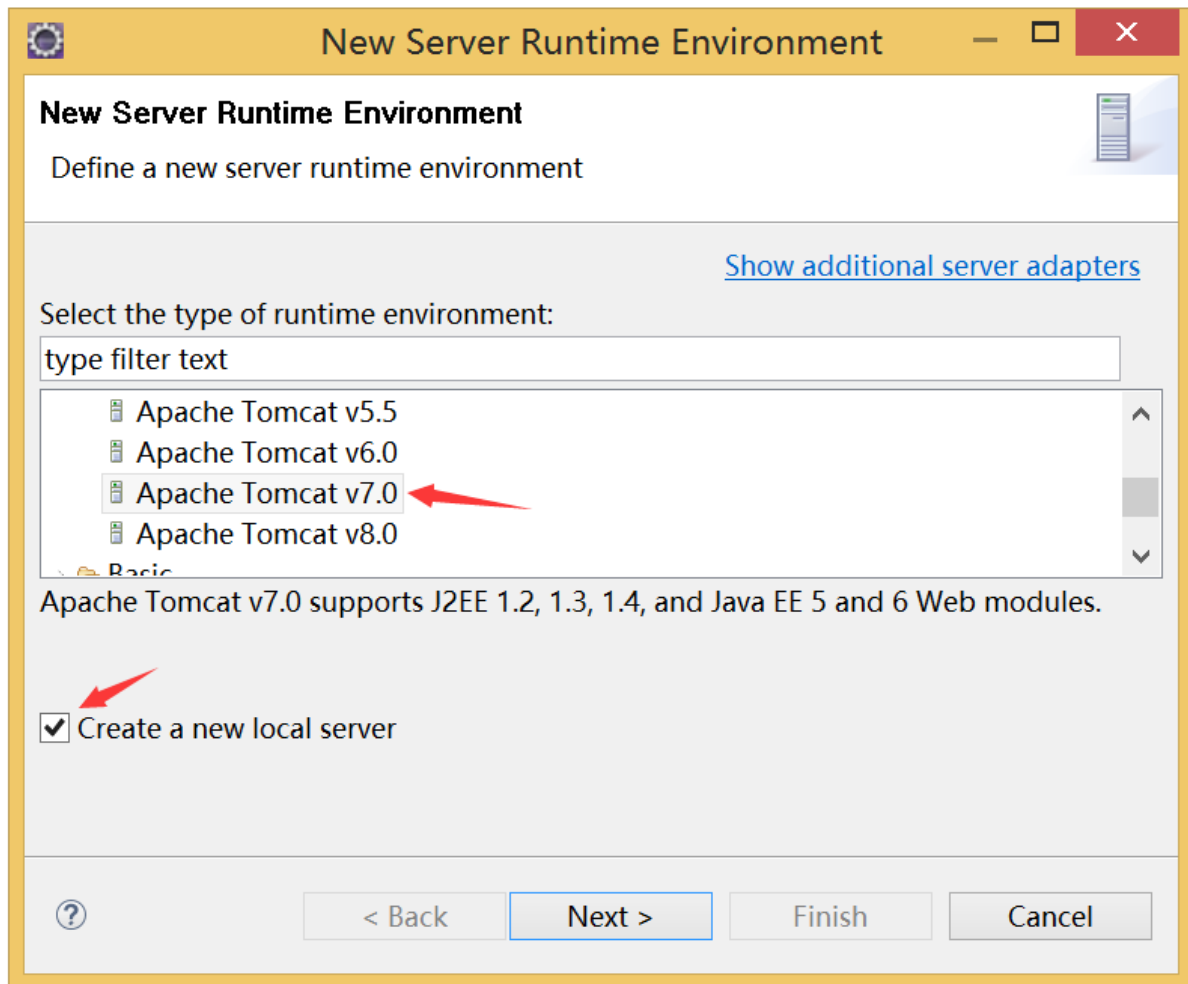


搭建服务器

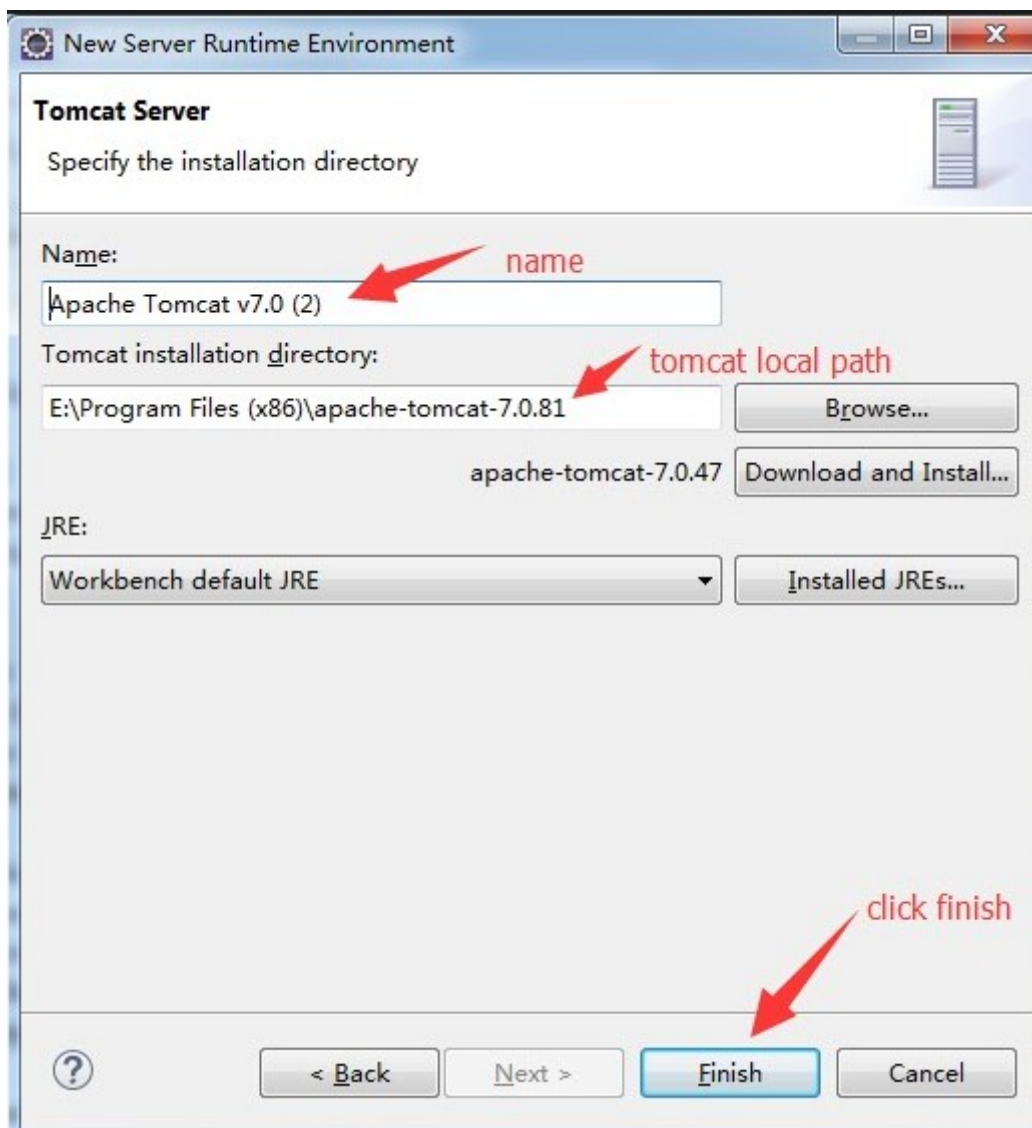
1. 下载Tomcat, 解压

2. 菜单栏 window 的 Preferences 选项

3. Preferences -> Server -> RuntimeEnvironments -> 点击右边的 add



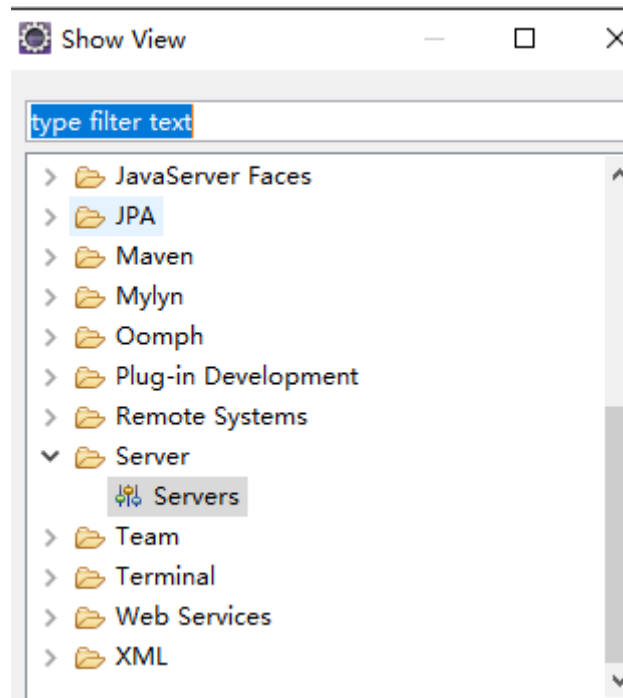
4. 加载tomcat



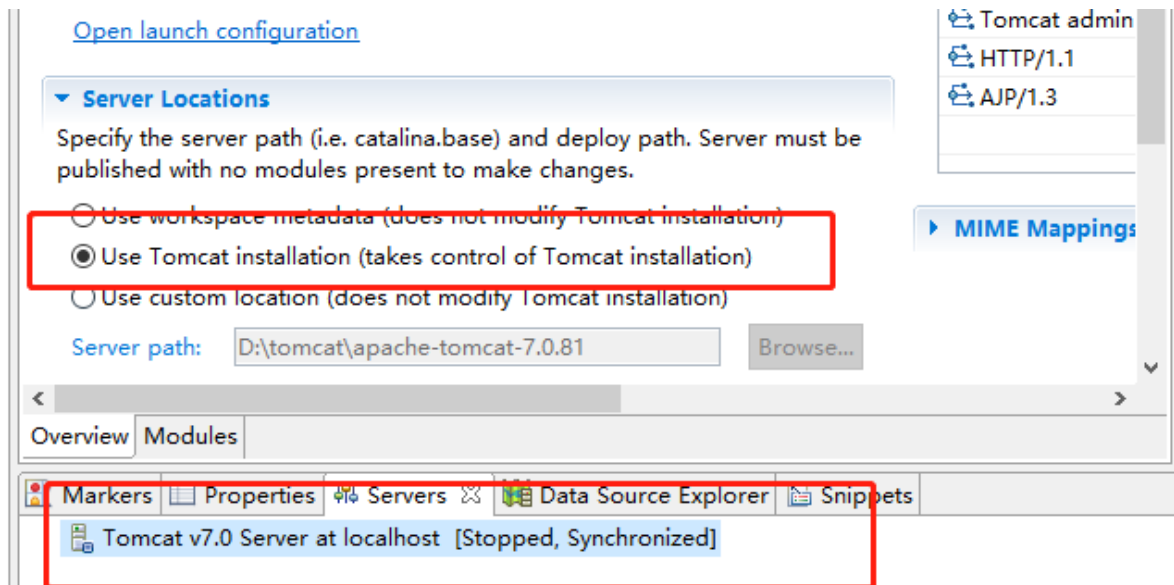
5.创建工程。右击 -> new -> Dynamic Web Project -> 请看下图

6, 显示server

window - show view - other - 选择server



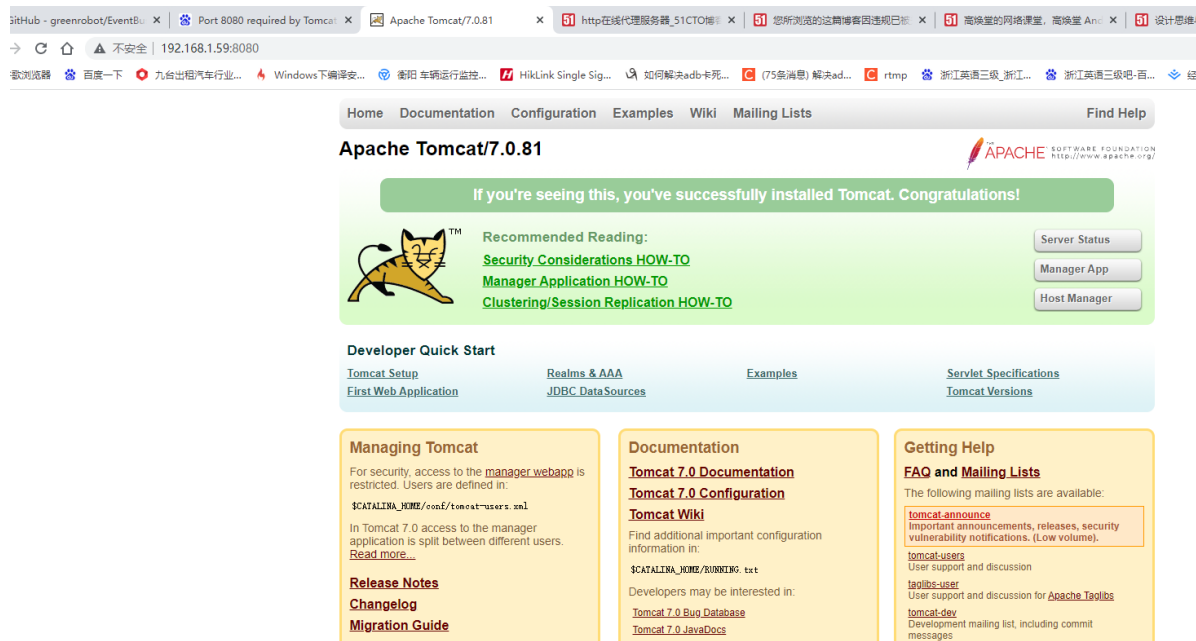
使用tomcat



7.测试发布 -> 请看下图 -> 右击tomcat -> 点击star

如果端口被占用，可以通过netstat -ano|findstr 8081 查看哪个进程占用的

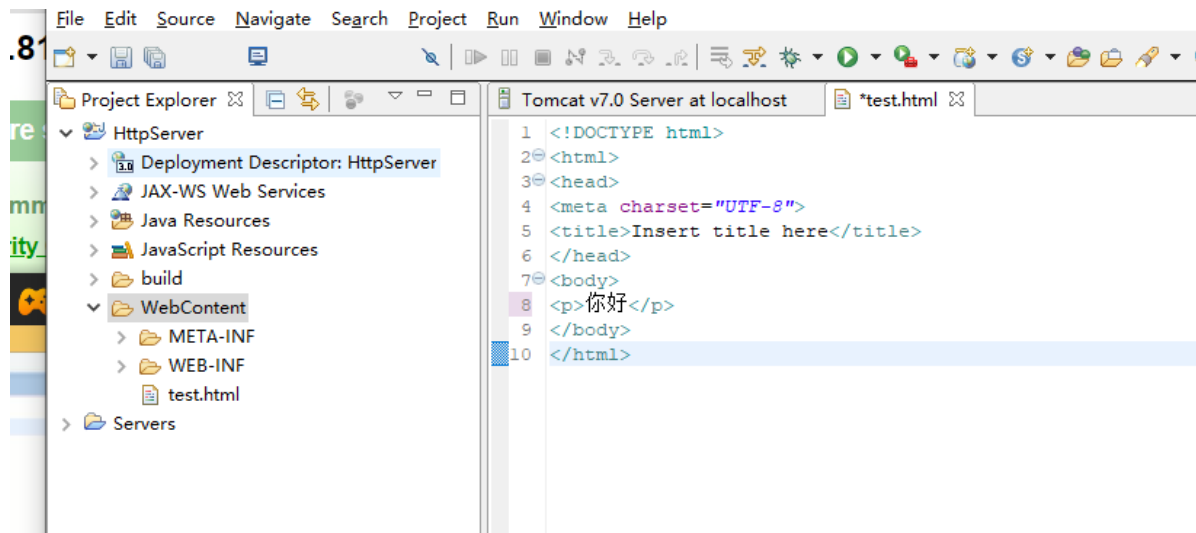
浏览器输入 ip: 8080 如果能看到tomcat 说明配置正确



tomcat默认8080端口

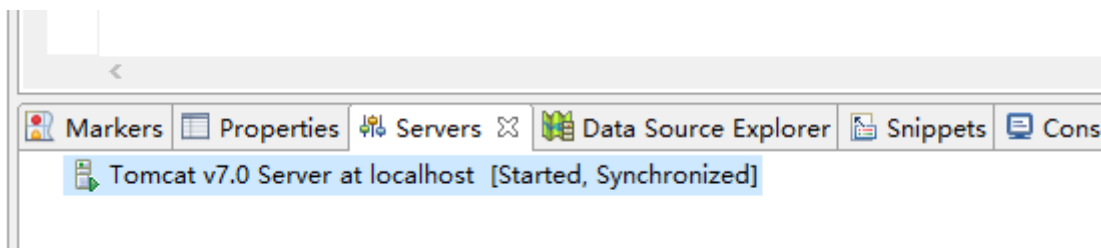
8.新建web工程

new 一个 Dynamic Web Project 工程。



在WebContent 根目录下新建一个html文件

9, 将新建的web工程发布到tomcat



右键 Add and Remove。添加我们刚刚新建的工程

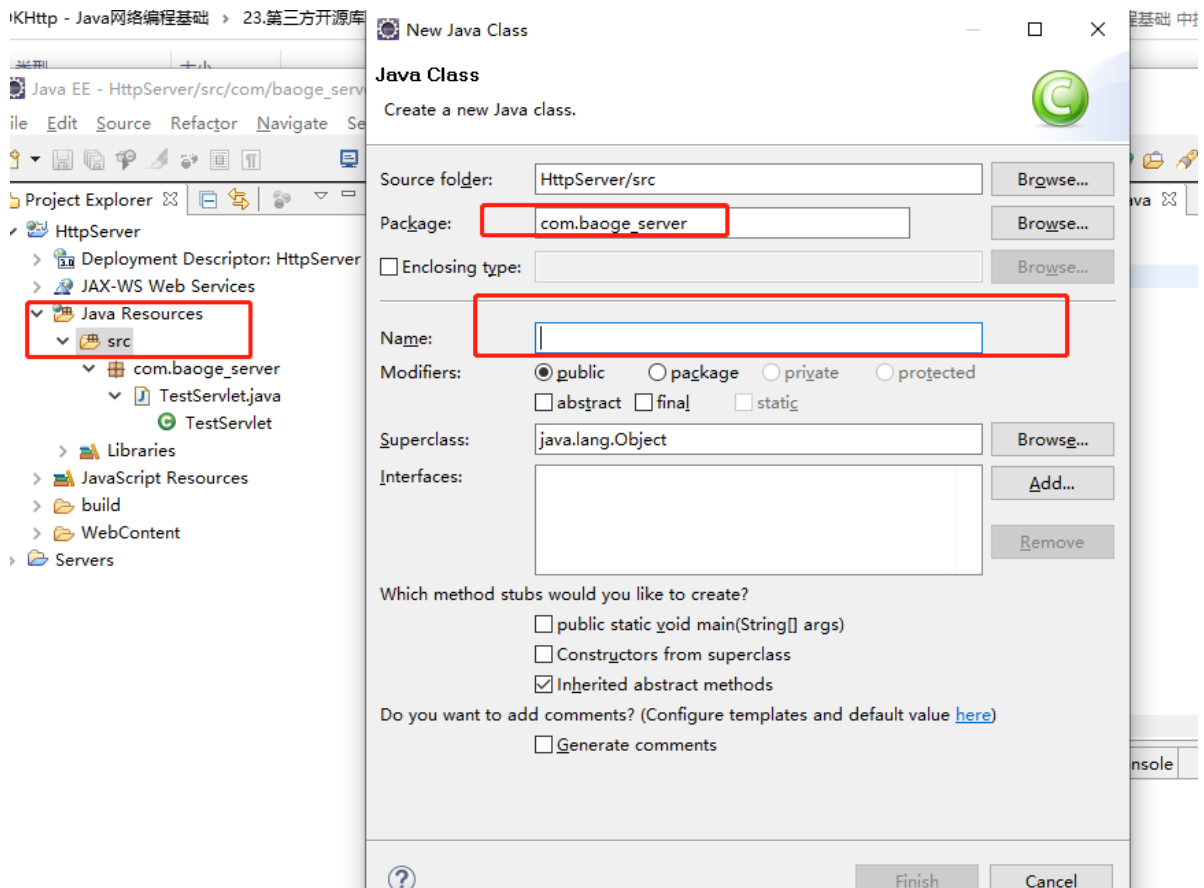
右键restart 进行发布

用浏览器打开网页

http://本机地址:8080/HttpServer/test.html

就能看到结果

测试servlet接口



```
1 package com.baoge_server;  
2  
3 import java.io.File;  
4 import java.io.IOException;  
5 import java.io.PrintWriter;  
6  
7 import javax.servlet.ServletException;
```

```

8  import javax.servlet.annotation.WebServlet;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13 @WebServlet("/TestServlet")
14 public class TestServlet extends HttpServlet{
15
16     private static final long serialVersionUID = 1L;
17
18     @Override
19     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
20         doPost(req, resp);
21     }
22
23     @Override
24     protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
25         PrintWriter printWriter = resp.getWriter();
26         System.out.println("Request");
27         printWriter.write("test post");
28         printWriter.flush();
29         printWriter.close();
30     }
31
32     @Override
33     protected long getLastModified(HttpServletRequest req) {
34         // 这个文本有没有更新过
35         File file = new File(req.getContextPath(), "index.html");
36         return file.lastModified();
37     }
38 }
39

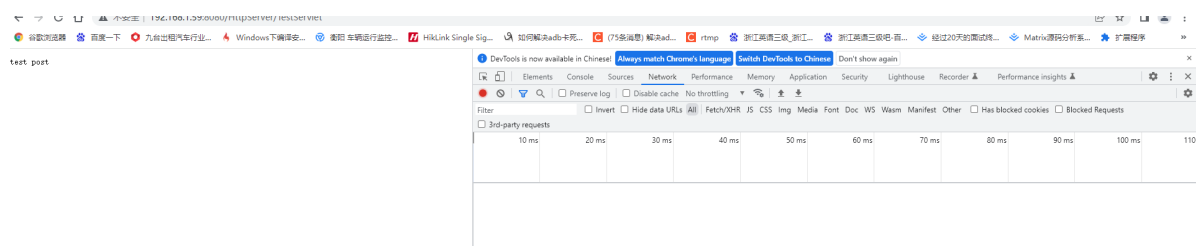
```

然后在浏览器中输入 <http://192.168.1.59:8080/HttpServer/TestServlet>

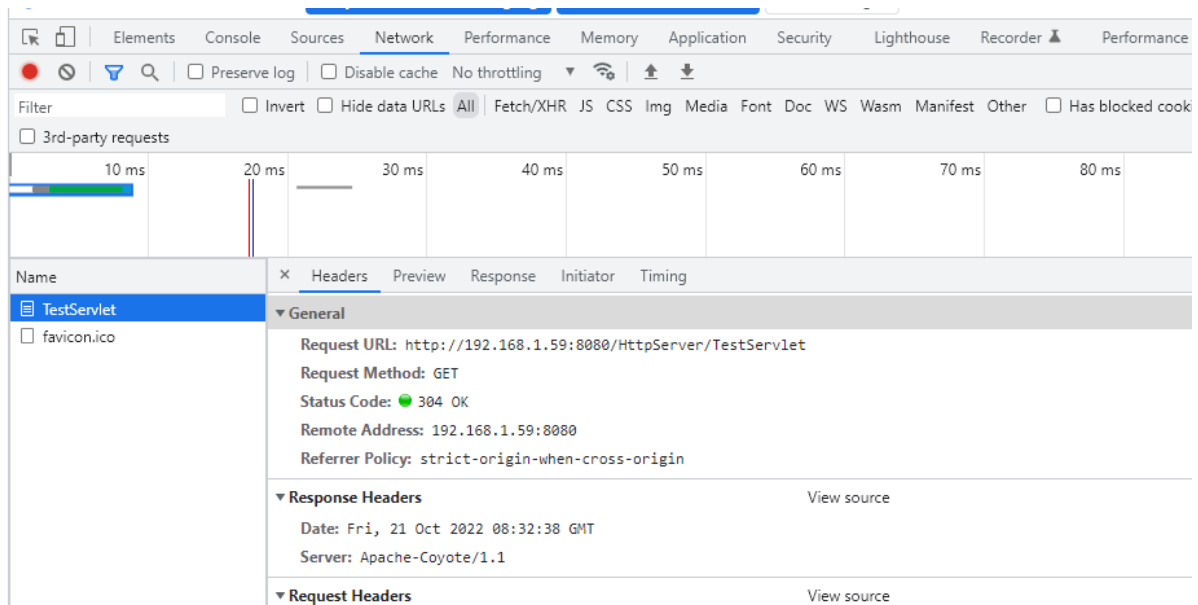
可以看到浏览器结果和tomcat打印



谷歌浏览器 网页上右键-检查 会弹出调试界面



再回车，可以看到报文



cookie

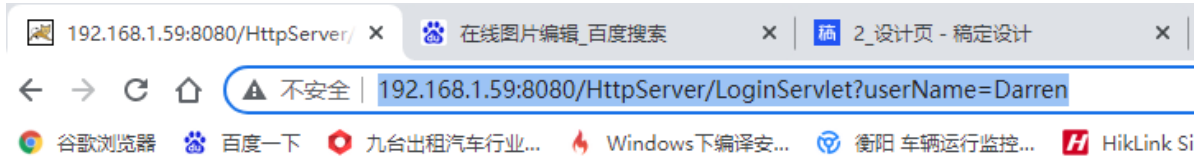
```
1 package com.baoge_server;
2
3 import javax.servlet.annotation.WebServlet;
4 import javax.servlet.http.Cookie;
5 import javax.servlet.http.HttpServletRequest;
6 import javax.servlet.http.HttpServletResponse;
7
8 import com.alibaba.fastjson.JSON;
9 import com.baoge_server.response.entity.ResponseEntity;
10 import com.baoge_server.response.entity.UserInfoEntity;
11 @WebServlet("/LoginServlet")
12 public class LoginServlet extends BaseJsonServlet{
13
14     private static final long serialVersionUID = 1L;
15
16     @Override
17     protected ResponseEntity onHandler(HttpServletRequest req, HttpServletResponse
18     resp) throws Exception {
19
20         ResponseEntity responseEntity = new ResponseEntity();
21         responseEntity.code = "0022";
22         responseEntity.msg = "用户名或密码错误";
23
24         String userName = req.getParameter("userName");
25         if("Darren".equals(userName)){
26             responseEntity.code = "0000";
27             responseEntity.msg = "登录成功";
28
29             UserInfoEntity infoEntity = new UserInfoEntity();
30             infoEntity.userName = userName;
31             infoEntity.userSex = "男";
32             responseEntity.data = JSON.toJSONString(infoEntity);
33             // 设置登录 Cookie 登录有效时间
34             Cookie cookie = new Cookie("userName", "Darren");
35             cookie.setMaxAge(10); // 设置Cookie的过期时间
```

```

36         resp.addCookie(cookie);
37     }
38
39
40     return responseEntity;
41 }
42
43 }
44

```

浏览器输入: <http://192.168.1.59:8080/HttpServer/LoginServlet?userName=Darren>



{\"code\":\"0000\", \"data\":{\"userName\":\"Darren\", \"userSex\":\"男\"}, \"msg\":\"登录成功\"}

服务端应答报文中会放回cookie, 客户端在**有效时间内**将返回的cookie放到访问报文中, 不在有效范围内则不带上cookie



使用效果可以看下面评论的例子, 用户评论的时候会判断cookie是否在有效期内, 如果在则评论成功, 否则失败

```

1  @WebServlet("/UserCommentServlet")
2  public class UserCommentServlet extends BaseJsonServlet{
3
4      private static final long serialVersionUID = 1L;
5
6      @Override
7      protected ResponseEntity onHandler(HttpServletRequest req, HttpServletResponse
resp) throws Exception {
8          // 判断该用户有没有登录, 没有登录或者登录过期都不让评论
9
10         ResponseEntity responseEntity = new ResponseEntity();
11         responseEntity.code = "0033";
12         responseEntity.msg = "评论失败, 用户未登录";
13
14         Cookie[] cookies = req.getCookies();
15         if(cookies!=null){
16             for(Cookie cookie : cookies){
17                 String userName = cookie.getName();
18                 if(userName != null){

```



```
19         responseEntity.code = "0000";
20         responseEntity.msg = "评论成功";
21         return responseEntity;
22     }
23 }
24 }
25
26     return responseEntity;
27 }
28
29 }
30
```

用户先登录 <http://192.168.1.59:8080/HttpServer/LoginServlet?userName=Darren> , 10秒内访问

<http://192.168.1.59:8080/HttpServer/UserCommentServlet> 则评论成功, 否则失败

Http缓存

Cache-Control (缓存策略) : Public、private、no-cache、max-age 、no-store (不缓存)

Expires (缓存的过期策略) : 指名了缓存数据有效的绝对时间, 告诉客户端到了这个时间点 (比照客户端时间点) 后本地缓存就作废了,

在这个时间点内客户端可以认为缓存数据有效, 可直接从缓存中加载展示。

<https://mp.weixin.qq.com/s/qOMO0LldA47j3RjhbCWUEQ>

如果有缓存并且过期了那么发起请求, 那么服务端会给我们数据? (不一定会给) 服务器的数据没有变动就不会给, 状态码会变为 304 , 自己拿之前过期的缓存